Scientific Excellence • Resource Protection & Conservation • Benefits for Canadians
Excellence scientifique • Protection et conservation des ressources • Bénéfices aux Canadiens

A Versatile Computer Program for Mixed Stock Fishery Composition Estimation

R. B. Millar

Science Branch
Department of Fisheries and Oceans
P. O. Box 5667
St. John's, Newfoundland A1C 5X1

September 1990

Canadian Technical Report of Fisheries and Aquatic Sciences No. 1753

Fisheries and Oceans Pêches et Océans

Canada

Canadian Technical Report of Fisheries and Aquatic Sciences

Technical reports contain scientific and technical information that contributes to existing knowledge but which is not normally appropriate for primary literature. Technical reports are directed primarily toward a worldwide audience and have an international distribution. No restriction is placed on subject matter and the series reflects the broad interests and policies of the Department of Fisheries and Oceans, namely, fisheries and aquatic sciences.

Technical reports may be cited as full publications. The correct citation appears above the abstract of each report. Each report is abstracted in *Aquatic Sciences and Fisheries Abstracts* and indexed in the Department's annual index to scientific and technical publications.

Numbers 1–456 in this series were issued as Technical Reports of the Fisheries Research Board of Canada. Numbers 457–714 were issued as Department of the Environment, Fisheries and Marine Service, Research and Development Directorate Technical Reports. Numbers 715–924 were issued as Department of Fisheries and the Environment, Fisheries and Marine Service Technical Reports. The current series name was changed with report number 925.

Technical reports are produced regionally but are numbered nationally. Requests for individual reports will be filled by the issuing establishment listed on the front cover and title page. Out-of-stock reports will be supplied for a fee by commercial agents.

Rapport technique canadien des sciences halieutiques et aquatiques

Les rapports techniques contiennent des renseignements scientifiques et techniques qui constituent une contribution aux connaissances actuelles, mais qui ne sont pas normalement appropriés pour la publication dans un journal scientifique. Les rapports techniques sont destinés essentiellement à un public international et ils sont distribués à cet échelon. Il n'y a aucune restriction quant au sujet; de fait, la série reflète la vaste gamme des intérêts et des politiques du ministère des Pêches et des Océans, c'est-à-dire les sciences halieutiques et aquatiques.

Les rapports techniques peuvent être cités comme des publications complètes. Le titre exact paraît au-dessus du résumé de chaque rapport. Les rapports techniques sont résumés dans la revue *Résumés des sciences aquatiques et halieutiques*, et ils sont classés dans l'index annual des publications scientifiques et techniques du Ministère.

Les numéros 1 à 456 de cette série ont été publiés à titre de rapports techniques de l'Office des recherches sur les pêcheries du Canada. Les numéros 457 à 714 sont parus à titre de rapports techniques de la Direction générale de la recherche et du développement, Service des pêches et de la mer, ministère de l'Environnement. Les numéros 715 à 924 ont été publiés à titre de rapports techniques du Service des pêches et de la mer, ministère des Pêches et de l'Environnement. Le nom actuel de la série a été établi lors de la parution du numéro 925.

Les rapports techniques sont produits à l'échelon régional, mais numérotés à l'échelon national. Les demandes de rapports seront satisfaites par l'établissement auteur dont le nom figure sur la couverture et la page du titre. Les rapports épuisés seront fournis contre rétribution par des agents commerciaux.

91-04665/

Canadian Technical Report of
Fisheries and Aquatic Sciences 1753

September 1990

A VERSATILE COMPUTER PROGRAM FOR MIXED STOCK FISHERY COMPOSITION ESTIMATION

by

R. B. Millar

Science Branch
Department of Fisheries and Oceans
P.O Box 5667
St John's, Newfoundland A1C 5X1
Canada

ABSTRACT

Millar, R. B. 1990. A versatile computer program for mixed stock fishery composition estimation. Can. Tech. Rep. Fish. Aquat. Sci. 1753: iii + 29 p.

This technical report describes a FORTRAN program (HISEA.FOR) to be used in the study of mixed stock fishery composition estimation. Program HISEA is a versatile research tool capable of performing analyses, The simplest use of HISEA is to perform a composition analysis. The program can then be run in bootstrap mode to provide a non-parametric estimate of the reliability of the estimated compositions. In simulation mode, HISEA can quantify the effect of changing sample sizes; of adding or removing variables; and of adding or removing stocks. Simulation also identifies potential problems (such as high bias and variance in the composition estimators) that may arise due to different stocks being too alike.

RÉSUMÉ

Millar, R. B. 1990. A versatile computer program for mixed stock fishery composition estimation. Can. Tech. Rep. Fish. Aquat. Sci. 1753: iii + 29 p.

Le présent rapport technique décrit l'application d'un programme FORTRAN (HISEA.FOR) à l'étude des estimations de composition des stocks divers de poisson. Le programme HISEA est un outil de recherche à usages multiples qui peut effectuer des analyses. L'emploi le plus simple de HISEA consiste à réaliser une analyse de composition. Le programme peut alors être exécuté en mode bootstrap pour fournir une estimation non paramétrique de la fiabilité des compositions estimées. En mode de simulation, HISEA peut quantifier les résultats de changements de tailles d'échantillons, de l'ajout ou du retrait de variables et de l'ajout ou du retrait de stocks. La même simulation identifiera les problèmes potentials (comme un biais élevé et une variance des estimateurs de composition) que pourrait provoquer une trop grande ressemblance entre stocks différents.

©Minister of Supply and Services Canada 1990
Cat. No. Fs 97-6/1753E ISSN 0706-6457

Correct citation for this publication:

Millar, R. B. 1990. A versatile computer program for mixed stock fishery composition estimation. Can. Tech. Rep. Fish. Aquat. Sci 1753: iii + 29 p.

INTRODUCTION

The motivation for creation of the simulation/bootstrap/analysis program HISEA was to perform comparison studies of the relative performance of several different estimators of mixed fishery composition. The output of HISEA (Appendix 3) includes five different estimates. These estimators are listed below - for more detail the reader is referred to Millar (in press). The notation and terminology used throughout this technical report is consistent with that of Millar (in press).

Using HISEA, Millar (in press) found the direct maximum likelihood estimator Θ_5 to be superior and encourages the use of it alone. However, since HISEA is a research program, the other four estimators continue to be calculated. When performing long simulation or bootstrap runs one might consider revising HISEA to calculate only Θ_5 if speed of execution becomes a problem.

The original version of HISEA was written in 1987 for the University of Washington's CDC Cyber mainframe computer. In 1990 it was revised to run under Digital's VMS operating system (Appendix 4).

PROGRAM SUMMARY

ESTIMATORS

The five estimators calculated by HISEA are

Θ_1 : Raw classification proportions
Θ_2 : Cook and Lord corrected classification estimator
Θ_3 : Cook's constrained corrected classification estimator
Θ_4 : Maximum classification-likelihood estimator
Θ_5 : Direct maximum likelihood estimator

The first four estimators are based on a classification step. Inadequacies in estimator Θ_1 (it is not a statistically consistent estimator) prompted the development of estimator Θ_2 by Worlund and Fredin (1962), Cook and Lord (1978), Pella and Robertson (1979) and Cook (1982). Cook (1983) presents Θ_3 . Millar (1987) focuses on Θ_5 but also discovered its relationship with Θ_2 and Θ_3 , and proposes Θ_4 as an alternative to Θ_2 and Θ_3 . Θ_5 is used by Milner (1981,1983) and Fournier (1984).

Estimator Θ_1 is not consistent, and estimator Θ_2 is not well defined since it can produce negative estimates of composition. Thus, of the five estimators, only Θ_3 , Θ_4 or Θ_5 are statistically valid. Millar (in press) recommends using Θ_5 since it has better performance (lower mean squared error) than Θ_3 and Θ_4 .

ASSUMPTIONS

Program HISEA assumes that the variables used are approximately distributed as multi-variate normal and that they have a covariance matrix that is common to all stocks. Variables taken from scales (circuli distances or counts, etc) will usually satisfy this assumption. Researchers using other types of variables (genotypes, etc) will have to modify HISEA accordingly (see below).

Array X contains likelihood values, that is, $X(i,j)$ is the likelihood of observing the measurements made on fish i given that it is from stock j. Under the assumption of multivariate normality, calculation of the log-likelihood is defined by a linear combination of the measured variables (the linear discriminant function). These coefficients are calculated in subroutine LDF and the likelihoods $X(i,j)$ are calculated in subroutine CLASFY (this subroutine then classifies the fish on the basis of these likelihoods - see Millar (in press)).

A researcher wishing to use variables that are not multi-variate normal, or do not satisfy the constant covariance matrix assumption, will have to modify subroutines LDF and CLASFY. The modification requires only that the likelihood values $X(i,j)$ are calculated according to the assumptions made concerning the distribution of the measured variables.

IMPLEMENTATION

Estimation of the classification matrix

Program HISEA estimates the classification matrix Φ by classifying the baseline data, see Millar (in press). (Note that when doing simulation or bootstrap runs without resampling from the given baseline, then the classification matrix is fixed throughout the run. If resampling the baseline then the classification matrix will most likely be different for every bootstrap or simulation in the run.) Some authors (e.g., Pella and Robertson 1979) split the baseline data in two, using one half to devise the classification rule and estimate Φ by classifying the second half. A third possibility is to use a leave-one-approach. The difference in these methods will be negligible in practice, provided the baseline is of a reasonable size, say, ≥ 100 fish in total.

The estimator of Φ used by HISEA is the simplest and is also the one used by the SAS discrimination procedure DISCRIM. In addition, when simulating without resampling the baseline, the estimated Φ is, in fact, the true classification matrix since the baseline defines the population being simulated (Millar, in press).

Maximization of Likelihood

Calculation of estimators Θ_4 and Θ_5 is a constrained non-linear maximization problem. There are several ways to perform this maximization – fortunately, the choice of method does *not* affect the estimators. One simply requires an algorithm that can be relied upon to converge to the maximum of the likelihood function.

Program HISEA uses the expectation-maximization (EM) algorithm (Dempster et al. 1977) within subroutine EM. The advantage of the EM algorithm is that, in this application, it is exceptionally easy to program and it can be written in a single line (Milner et al. 1983; Millar 1987). The algorithm is extremely robust and it never fails to converge to the maximum. The EM algorithm used by HISEA has been accelerated (subroutine ACCEL) and this speeds up the algorithm when the estimates from successive iterations are close to converging. The maximum number of iterations has been set to 100 (integer variable IMAX) and the algorithm is deemed to have converged if the maximum absolute difference between successive estimates of composition is less than 10^{-6} (real variable TOL).

INSTRUCTIONS FOR RUNNING HISEA

The program requires 3 input files for analysis or bootstrap runs, the baseline data (data on fish of known origin), a control file with parameters and options, and a file with the mixed sample (unknown origin) data. A simulation run does not need the later file since it generates its own mixed sample data. The default names for the input files are FOR007.DAT, FOR008.DAT and FOR009.DAT respectively. Examples of FOR007.DAT and FOR008.DAT are given in Appendices 1 and 2 respectively. The output produced by running HISEA with these input files is given in Appendix 3.

Below is an example of the control file FOR008.DAT, followed by an explanation of the records on each line.

```
Line 1   '1985 Atlantic salmon trial run'
Line 2   'SIMULATION'
Line 3   4   'Europe'   'Canada'   'USA'   'Greenlan'
Line 4   9
Line 5   'STD'   'Y'
Line 6   500   500   500   500
Line 7   'MIX'   Y'
Line 8   84357
Line 9   10
Line 10   100   0.00   0.00   0.00   1.00
```

Line 1: Title; ≤ 80 characters.

Line 2: Type of run (Simulation, analysis, or bootstrap).

Line 3: Number and names of stock groups; number must be ≤ 8; only the first 8 characters of the names are used.

Line 4: Number of variables (≤ 16).

Line 5: Whether or not to resample the baseline. This line must read either 'STD' 'N' or 'STD' 'Y', corresponding to no resampling and resampling respectively. (Here STD is an abbreviation for "standard", which is used instead of "baseline" by some researchers.)

Line 6: Size desired for resampled baselines; can be any size (sampling with replacement); must be positive; must include this line even if not resampling (option 'N' on line 5).

Line 7: Whether or not to resample/simulate the mixed sample. This line must read either 'MIX' 'N' or 'MIX' 'Y' . If simulating, the mixed sample is generated from the baseline data (FOR007.DAT) and 'MIX' 'Y' must be specified.

Line 8: Seed for random number generator; must be a (large) positive integer.

Line 9: Number of simulations desired; must be between 1 and 1000.

Line 10: Size of mixed sample to be taken from a simulated mixed fishery population with composition given by the remaining values on this line (this line read only for simulations).

All lines are freefield format. Only the first four lines are read for an Analysis run.

The baseline and mixed sample files are also freefield format and consist of variable values only, with no header or other identifier information on the line. The mixed sample file FOR009.DAT contains variable values for fish from a sample of the mixed population. This file is not required if HISEA is used in simulation mode. In the version of HISEA presented here, the baseline and mixed sample files must not exceed 1400 lines each.

The baseline file (FOR007.DATA) contains the baseline data for all of the stock groups, linked end-to-end and separated by the end-of-file marker. (On the Northwest Atlantic Fisheries Centre's VAX 6310 this is the control Z character and it can be inserted with the edt editor by hitting the <PF1> key followed by the 3 on the extended keyboard. At the command line prompt type 26 (from the regular keyboard) and hit the <Do> key. If done correctly the character will appear as ^Z and will be inserted at the current position of the cursor.) Each line in FOR007.DAT represents the measurements made on a fish of known origin. The stocks are assumed to be in the order as given in line 3 above. The program automatically determines the size of the baseline (i.e., number of fish) for each stock by using the ^Z markers.

PORTABILITY

Program HISEA has also been ported to SUN Microsystems workstations. The intrinsic functions that appear not be compatible across all implementations of FORTRAN are DATE and TIME (subroutine READ8) and the random number generator RAN (subroutine ORDVEC). The code in subroutine READIN may need to be modified if the end-of-file marker cannot be inserted in file FOR007.DAT.

REFERENCES

Cook, R. C., and G. E. Lord. 1978. Identification of stocks of Bristol Bay sockeye salmon, *Oncorhynchus nerka,* by evaluating scale patterns with a polynomial discriminant method. Fish. Bull. 76: 415-423

Cook, R. C. 1982. Stock identification of sockeye salmon *(Oncorhynchus nerka)* with scale pattern recognition. Can. J. Fish. Aquat. Sci. 39: 611-617.

Cook, R. C. 1983. Simulation and application of stock composition estimators. Can. J. Fish. Aquat. Sci. 40: 2113-2118

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. B39: 1-38.

Fournier, D. A., T. D. Beacham, B. E. Riddell, and C. A. Busack. 1984. Estimating stock composition in mixed stock fisheries using morphometric, meristic, and electrophoretic characteristics. Can. J. Fish. Aquat. Sci. 41: 400-408.

Millar, R. B. 1987. Maximum likelihood estimation of mixed stock fishery composition. Can. J. Fish. Aquat. Sci. 44: 583-590

Millar, R. B. In press. Comparison of methods for estimating mixed stock fishery composition. Can. J. Fish. Aquat. Sci.

Milner, G. B., D. J. Teel and F. M. Utter. 1983. Genetic stock identification study. Annual report of research. NWAFC, NOAA, Seattle, WA 98112.

Milner, G. B., D. J. Teel, F. M. Utter, and C.L. Burley. 1981. Columbia River stock identification study: validation of genetic method. Annual report of research (FY80). NWAFC, NOAA, Seattle, WA 98112.

Pella, J. J., and T. L. Robertson. 1979. Assessment of composition of stock mixtures. Fish. Bull. 77: 387-398.

Worlund, D. D., and R. A. Fredin. 1962. Differentiation of stocks, p. 143-153. *In* Symposium on pink salmon. H. R. MacMillan Lectures in Fisheries, University of British Columbia, Vanouver, B.C.

APPENDIX 1

This is a sample baseline data file (FOR007.DAT) with 32 and 24 fish respectively. The data are circuli counts and are a subset of a 1985 baseline for European and North American atlantic salmon.

```
26 2
22 7
24 7
27 5
26 4
26 2
33 2
29 6
25 4
21 5
33 4
24 5
32 3
29 5
29 5
23 5
23 4
28 2
27 5
25 4
30 3
25 2
21 4
30 3
26 3
29 2
29 2
27 3
34 2
29 5
29 6
32 3
 2
18 10
32 3
24 2
26 5
27 7
24 3
27 2
30 5
19 9
12 4
20 7
14 4
16 4
19 3
13 9
23 6
21 5
24 5
19 8
25 8
26 5
22 13
28 7
23 6
```

APPENDIX 2

A sample control file (FOR008.DAT) for performing a simulation using the baseline data from Appendix 1. Note that the resampled baselines are specified to be size 50 and that the mixed sample size is 200. HISEA will perform 100 simulations from a hypothetical mixed fishery with equal contribution of European and North American fish.

```
'1985 Atlantic salmon trial run'
'SIMULATION'
2 'Europe' 'Nth America'
2
'STD' 'Y'
50 50
'MIX' 'Y'
123456
100
200 0.9 0.1
```

APPENDIX 3

Listing from simulation run using the input files listed in Appendices 1 and 2. The maximum classification-likelihood estimator and direct maximum likelihood estimator are labelled the "Millar constrained" and "maximimum likelihood" estimators respectively. This run took 8 seconds of CPU time on a VAX 6310.

```
$run hisea

PROGRAM  HISEA.......EXECUTION DATE:  9-APR-90      14:04:30

1985 Atlantic salmon trial run


FUNCTION OF THIS RUN IS.........SIMULATION
#STOCKS IN THE MODEL.............. 2
THE STOCKS ARE..................Europe   Nth Amer
#VARIABLES USED................... 2

STANDARD BEING RESAMPLED?........Y
RESAMPLED STANDARD SIZES.........    50     50
MIXTURE BEING SIMULATED?.........Y

RANDOM NUMBER GENERATOR SEED....  123456
NUMBER OF RUNS REQUESTED?.......  100
SIZE OF SIMULATED MIXTURE.......  200
ACTUAL COMPOSITION IS...........   0.900   0.100
```

==========

MEAN AND STD DEV SUMMARY OF VARIABLES

VAR	Europe	Nth Amer
1	27.281 (3.49)	22.167 (5.25)
2	3.8750 (1.52)	5.8333 (2.71)

==========

THE SIZES OF THE STANDARDS ARE 32 24

TABLE OF COMPOSITION ESTIMATE MEANS. NUMBER OF RUNS = 100

	RAW	COOK & LORD	COOK CONSTRAINED	MILLAR CONSTRAINED	MAXIMUM LIKELIHOOD
Europe	0.7420	0.9172	0.8914	0.8913	0.9470
Nth Amer	0.2580	0.0828	0.1086	0.1087	0.0530

TABLE OF COMPOSITION ESTIMATE STANDARD DEVIATIONS OVER THE 100 RUNS

	RAW	COOK & LORD	COOK CONSTRAINED	MILLAR CONSTRAINED	MAXIMUM LIKELIHOOD
Europe	0.0359	0.1406	0.0972	0.0971	0.0489
Nth Amer	0.0359	0.1406	0.0972	0.0971	0.0489

TABLE OF SQRT OF MEAN SQUARED ERRORS

	RAW	COOK & LORD	COOK CONSTRAINED	MILLAR CONSTRAINED	MAXIMUM LIKELIHOOD
Europe	0.1620	0.1416	0.0976	0.0975	0.0678
Nth Amer	0.1620	0.1416	0.0976	0.0975	0.0678

THE COVARIANCE MATRIX OF THE 100 MAXIMUM LIKELIHOOD COMPOSITION ESTIMATES IS:

(POPULATIONS 1- 2 * 1- 2)

	1	2
1	0.2395E-02	
2	-0.2394E-02	0.2394E-02

THE CORRESPONDING CORRELATION MATRIX IS:

(POPULATIONS 1- 2 * 1- 2)

	1	2
1	0.1000E+01	
2	-0.9999E+00	0.1000E+01

FORTRAN STOP
KIWI job terminated at 9-APR-1990 14:04:39.39

Accounting information:

Buffered I/O count:	83	Peak working set size:	490
Direct I/O count:	64	Peak page file size:	3003
Page faults:	796	Mounted volumes:	0
Charged CPU time:	0 00:00:08.02	Elapsed time:	0 00:00:11.96

APPENDIX 4 – PROGRAM HISEA

```
      PROGRAM HISEA
C
C 16 OCT 87: THIS PROGRAM WAS WRITTEN BY  RUSSELL MILLAR
C            (DEPT OF STATISTICS, GN-22, UNIVERSITY OF WASHINGTON,
C            SEATTLE) FOR THE FISHERIES RESEARCH INSTITUTE,
C            UNIVERSITY OF WASHINGTON.
C
C 13 MAR 90: MODIFIED  BY  RUSSELL MILLAR  (CANADIAN DEPT OF FISHERIES
C            AND OCEAN, ST JOHN'S, NFLD) TO RUN UNDER THE VMS
C            OPERATING SYSTEM.
C
C.....
C FOR007.DAT CONTAINS ALL OF THE STANDARDS APPENDED TOGETHER WITH
C   CONTROL Z CHARACTERS SEPARATING THEM.
C FOR008.DAT IS THE CONTROL FILE - SPECIFIES PURPOSE OF RUN (ANALYSIS,
C   SIMULATION OR BOOTSTRAP) AND #POPNS, #VARIABLES   ETC ETC.
C FOR009.DAT CONTAINS THE MIXED SAMPLE FILE.  IT IS NOT REQUIRED BY A
C   SIMULATION RUN.
C.....
C SOME VARIABLES USED INCLUDE:
C   NP=NUMBER OF STOCK GROUPS (POPULATIONS) ON FOR007.DAT
C   NV=NUMBER OF VARIABLES
C   NSTD=TOTAL NUMBER OF FISH OVER ALL THE STANDARDS
C   NMIX=SIZE OF MIXED SAMPLE ON FOR009.DAT
C   N=SIZE OF SIMULATED MIXED SAMPLE
C*************************************************************************
C***********PRESET LIMITS FOR THIS VERSION OF SOURCE CODE ARE**********
C*********** NP<=8   NV<=16  NSTD<=1400   NMIX<=1400   N<=2000 ********
C*************************************************************************
C THE INDEX VARIABLES IN  DO  LOOPS ARE PREDOMINANTLY USED IN THE
C   FOLLOWING WAY:
C   J  INDEXS THE STOCK GROUP    J=1,...,NP
C   G  INDEXS FISH IN THE STDS OR MIXED SAMPLE    G=1,...,NSTD OR NMIX
C   L  INDEXS THE VARIABLES    L=1,...,NV
C SOME ARRAYS USED INCLUDE:
C   STSIZE(J)=SIZE OF STANDARD FOR STOCK GROUP J
C   GSIZE(J)=SIZE OF RESAMPLED STANDARD.  OFTEN THE SAME AS STSIZE(J).
C   FREQ(G)=FREQUENCY OF FISH  G  IN THE RESAMPLED DATA.
C   CLASS(G)=THE CLASS FISH  G  IS CLASSIFIED TO.
C   VARBLE(G,L)=THE L'TH MEASUREMENT ON THE G'TH FISH.
C   THETA(J,K)=THE CONTRIBUTION ESTIMATES
C            THETA(*,1)=RAW CLASSIFICATION ESTIMATES
C            THETA(*,2)=COOK AND LORD CORRECTED ESTIMATES
C            THETA(*,3)=COOK CONSTRAINED CORRECTED ESTIMATES
C            THETA(*,4)=MILLAR CONSTRAINED CORRECTED ESTIMATES
C             NB: THE LAST THREE TYPES OF ESTIMATES WILL BE IDENTICAL
C                 WHEN THE COOK AND LORD CORRECTED ESTIMATES ARE ALL
C                 NON NEGATIVE.
C            THETA(*,5)=DIRECT MAXIMUM LIKELIHOOD ESTIMATES
C   ACTUAL()=TRUE VECTOR OF CONTRIBUTIONS FOR SIMULATION PURPOSES
C*************************************************************************
C THE MAIN PROGRAM READS THE CONTROL FILE (FOR008.DAT) AND CALLS THE
C APPROPRIATE SUBPROGRAMS DEPENDING ON THE PURPOSE OF THE RUN.  IT FIRST
C DISTINGUISHES CASES ACCORDING TO WHETHER THE STANDARDS ARE BEING
C RESAMPLED.
C*************************************************************************
C.....
      INTEGER NP,N,NV,NSAMPS,GSIZE(8)
      REAL ACTUAL(8),X(1400,8),STDVAR(1400,16),ORDERS(2001)
      REAL THSUM(8,5),THSSQ(8,5),COVAR(8,8)
      CHARACTER TYPE*10,OPT(2),POPNAM(8)*8
      INTEGER SEED
```

```
      COMMON THSUM,THSSQ,COVAR
      COMMON /UNIFORM/ ORDERS
      COMMON /OPTIONS/ TYPE,OPT
      COMMON /RANDOM/ SEED
C     CALL CONNEC(5)
      DO 1 J=1,8
        DO 2 K=1,5
          THSUM(J,K)=0.0
          THSSQ(J,K)=0.0
    2   CONTINUE
        DO 3 K=1,8
          COVAR(J,K)=0.0
    3   CONTINUE
    1 CONTINUE
      CALL READ8(NP,POPNAM,NV,N,NSAMPS,ACTUAL,GSIZE,SEED)
      IF(OPT(1).EQ.'N') THEN
C.....STANDARD NOT BEING RESAMPLED
        CALL BASEFIX(NP,NV,N,NSAMPS,ACTUAL,X,STDVAR,POPNAM)
      ELSE
        CALL NUBASES(NP,NV,N,NSAMPS,GSIZE,ACTUAL,X,STDVAR,POPNAM)
      END IF
      CALL RESULTS(NP,NSAMPS,ACTUAL,TYPE,POPNAM)
      STOP
      END
C*************************************************************************
      SUBROUTINE READ8(NP,POPNAM,NV,N,NSAMPS,ACTUAL,GSIZE,SEED)
C THIS SUBROUTINE READS THE CONTROL FILE (FOR008.DAT)
      INTEGER NP,NV,N,NSAMPS,GSIZE(8)
      REAL ACTUAL(8),ACTSUM
      INTEGER SEED
      CHARACTER TYPE*10,STD*3,MIX*3,OPT(2),POPNAM(8)*8,TITLE*80
      CHARACTER *10 YMD,HOUR
      COMMON /OPTIONS/ TYPE,OPT

      CALL DATE(YMD)
      CALL TIME(HOUR)
      WRITE(6,777) YMD,HOUR

      READ(8,*) TITLE
      WRITE(6,720) TITLE
      READ(8,*) TYPE
      WRITE(6,721) TYPE
      IF(TYPE(1:1).NE.'A'.AND.TYPE(1:1).NE.'B'.AND.TYPE(1:1).NE.'S')
     .  THEN
        PRINT *,'LINE 2 OF CONTROL FILE MUST SPECIFY  ANALYSIS ',
     .          'BOOTSTRAP OR SIMULATION'
        STOP
      END IF
      READ(8,*) NP,(POPNAM(J),J=1,NP)
      WRITE(6,722) NP
      WRITE(6,7225) (POPNAM(J),J=1,NP)
      IF(NP.LT.2.OR.NP.GT.8) THEN
        PRINT *,'NUMBER OF STOCKS IS OUT OF BOUNDS.  2 - 8'
        STOP
      END IF
      READ(8,*) NV
      WRITE(6,723) NV
      IF(NV.LT.1.OR.NV.GT.16) THEN
        PRINT *,'NUMBER OF VARIABLES IS OUT OF BOUNDS,  1 - 16'
        STOP
      END IF
      IF(TYPE(1:1).EQ.'A') THEN
        OPT(1)='N'
        OPT(2)='N'
```

```
      RETURN
      END IF
      READ(8,*) STD,OPT(1)
      WRITE(6,701) OPT(1)
      IF(STD.NE.'STD'.OR..NOT.(OPT(1).EQ.'Y'.OR.OPT(1).EQ.'N'))THEN
        PRINT *,'INVALID INPUT ON  STD LINE OF FOR008.DAT - CONSULT',
     .          ' PROGRAM DOCUMENTATION'
        STOP
      END IF
      READ(8,*) (GSIZE(J),J=1,NP)
      IF(OPT(1).EQ.'Y') THEN
        DO 8 J=1,NP
          IF(GSIZE(J).LT.1.0) THEN
            PRINT *,'SIZE OF RESAMPLED STANDARD FOR STOCK ',J,' IS NOT'
            PRINT *,'POSITIVE.   PROGRAM STOPPED'
            STOP
          END IF
    8   CONTINUE
        WRITE(6,7014) (GSIZE(J),J=1,NP)
      END IF
      READ(8,*) MIX,OPT(2)
      IF(MIX.NE.'MIX'.OR..NOT.(OPT(2).EQ.'Y'.OR.OPT(2).EQ.'N'))THEN
        PRINT *,'INVALID INPUT ON  MIX LINE OF FOR008.DAT - CONSULT',
     .          ' PROGRAM DOCUMENTATION'
        STOP
      END IF
      IF(TYPE(1:1).EQ.'B') THEN
        WRITE(6,7015) OPT(2)
      ELSE
        WRITE(6,7016) OPT(2)
      END IF
      IF(TYPE(1:1).EQ.'S'.AND.OPT(2).EQ.'N') THEN
        WRITE(6,702)
        STOP
      END IF
      IF(OPT(1).EQ.'N'.AND.OPT(2).EQ.'N') THEN
        WRITE(6,706) TYPE
        STOP
      END IF
      READ(8,*) SEED
      IF(SEED.LT.1000) THEN
        PRINT *,'SEED VALUE ',SEED,' IS TOO SMALL (LESS THAN 1000)'
        SEED=12345
        PRINT *,'SEED SET TO ',SEED
      END IF
      WRITE(6,705) SEED
      READ(8,*) NSAMPS
      WRITE(6,703) NSAMPS
      IF(NSAMPS.LT.1.OR.NSAMPS.GT.1000) THEN
        PRINT *,'NUMBER OF RESAMPLES REQUESTED IS OUT OF BOUNDS'
        STOP
      END IF
      IF(TYPE(1:1).EQ.'B') RETURN
      READ(8,*) N,(ACTUAL(J),J=1,NP)
      WRITE(6,704) N,(ACTUAL(J),J=1,NP)
      IF(N.LT.1) THEN
        PRINT *,'SIZE OF SIMULATED MIXTURE IS INVALID'
        STOP
      END IF
      ACTSUM=0.0
      DO 1 J=1,NP
        IF(ACTUAL(J).LT.0.0) THEN
          PRINT *,'NEGATIVE CONTRIBUTION GIVEN IN FOR008.DAT????????'
          STOP
        END IF
```

```
        ACTSUM=ACTSUM+ACTUAL(J)
    1 CONTINUE
      IF(ACTSUM.LT.0.999.OR.ACTSUM.GT.1.001) THEN
        PRINT *,'TOTAL CONTRIBUTION IS ',ACTSUM,'<>1.000??????'
        STOP
      END IF
      RETURN
  777 FORMAT(' PROGRAM  HISEA.......EXECUTION DATE: ',A10,4X,A10)
  720 FORMAT(/1X,A80//)
  721 FORMAT(1X,'FUNCTION OF THIS RUN IS.........',A10)
  722 FORMAT(1X,'#STOCKS IN THE MODEL............',I2)
 7225 FORMAT(1X,'THE STOCKS ARE.................',8(A8,1X))
  723 FORMAT(1X,'#VARIABLES USED.................',I2/)
  701 FORMAT(1X,'STANDARD BEING RESAMPLED?.......',A1)
 7014 FORMAT(1X,'RESAMPLED STANDARD SIZES........',8I6)
 7015 FORMAT(1X,'MIXTURE BEING RESAMPLED?........',A1/)
 7016 FORMAT(1X,'MIXTURE BEING SIMULATED?........',A1/)
  702 FORMAT(1X,'MUST GENERATE MIXTURE WHEN SIMULATING - EXPECTED ',
     .       'TO SEE ''MIX''  ''Y''  ON LINE 7 OF FOR008.DAT')
  703 FORMAT(1X,'NUMBER OF RUNS REQUESTED?.......',I5)
  704 FORMAT(1X,'SIZE OF SIMULATED MIXTURE.......',I5/,
     .       1X,'ACTUAL COMPOSITION IS...........',8F8.3)
  705 FORMAT(1X,'RANDOM NUMBER GENERATOR SEED....',I8)
  706 FORMAT(1X,A10,' IS IMPOSSIBLE SINCE NEITHER STANDARDS OR MIXTURE'/
     .       'ARE BEING RESAMPLED OR SIMULATED.')
      END
C*********************************************************************
      SUBROUTINE BASEFIX(NP,NV,N,NSAMPS,ACTUAL,X,STDVAR,PN)
C IF THIS ROUTINE CALLED THEN THE STANDARDS ARE NOT BEING RESAMPLED.
C NOTE: THE MATRIX OF MEASUREMENT LIKELIHOODS  X  CORRESPONDING TO THE
C STANDARD IS CALCULATED IN CLASFY (CALLED FROM STDUP).  IT WILL
C BE OVERWRITTEN IF DOING AN ANALYSIS OR BOOTSTRAP.
      INTEGER NP,NV,N,NMIX,NSAMPS,NSTD,STSIZE(8),CLASS(1400),G,SEVEN
      INTEGER FREQ(1400),GSIZE(8),NINE,ONE,DUMMY(8)
      REAL ACTUAL(NP),COEFS(8,17),PHI(8,8),X(1400,8),STDVAR(1400,16)
      CHARACTER TYPE*10,PN(NP)*6
      COMMON /OPTIONS/ TYPE
      SEVEN=7
      NINE=9
      ONE=1
      CALL READIN(NP,NV,STSIZE,NSTD,STDVAR,SEVEN,PN)
      WRITE(6,700) (STSIZE(I),I=1,NP)
      DO 800 G=1,NSTD
  800   FREQ(G)=1
      DO 801 J=1,NP
  801   GSIZE(J)=STSIZE(J)
      CALL STDUP(NP,NV,NSTD,STSIZE,GSIZE,FREQ,COEFS,CLASS,PHI,X,STDVAR)
      IF(TYPE(1:1).EQ.'S') THEN
        CALL SIMLATE(NP,N,NSAMPS,ACTUAL,NSTD,STSIZE,CLASS,PHI,X)
      ELSE
        CALL READIN(ONE,NV,DUMMY,NMIX,STDVAR,NINE,PN)
        WRITE(6,701) NMIX
        DO 802 G=1,NMIX
  802     FREQ(G)=1
        CALL CLASFY(NP,NV,NMIX,FREQ,CLASS,COEFS,STDVAR,X)
        CALL REALDAT(NP,NSAMPS,NMIX,CLASS,PHI,X)
      END IF
      RETURN
  700 FORMAT(//1X,'THE SIZES OF THE STANDARDS ARE ',8I6)
  701 FORMAT(/1X,'THE MIXED SAMPLE HAS ',I5,' FISH')
      END
C*********************************************************************
```

```
      SUBROUTINE NUBASES(NP,NV,N,NSAMPS,GSIZE,ACTUAL,X,STDVAR,PN)
C IF THIS ROUTINE IS CALLED THEN THE STANDARDS ARE BEING RESAMPLED.
      INTEGER NP,NV,N,NSAMPS,NSTD,STSIZE(8),CLASS(1400),BFREQ(1400),G,
     .        NINE,SEVEN,ONE,LOOP,DUMMY(8),MFREQ(1400),GSIZE(NP)
      REAL ACTUAL(NP),COEFS(8,17),PHI(8,8),X(1400,8),STDVAR(1400,16),
     .     MIXVAR(1400,16)
      CHARACTER TYPE*10,PN(NP)*6
      COMMON /OPTIONS/ TYPE
      NINE=9
      SEVEN=7
      ONE=1
      CALL READIN(NP,NV,STSIZE,NSTD,STDVAR,SEVEN,PN)
      WRITE(6,700) (STSIZE(J),J=1,NP)
      IF(TYPE(1:1).NE.'S') THEN
        CALL READIN(ONE,NV,DUMMY,NMIX,MIXVAR,NINE,PN)
        WRITE(6,709) NMIX
      END IF
      DO 1 LOOP=1,NSAMPS
        CALL RESAMP(NP,NSTD,GSIZE,STSIZE,BFREQ)
        CALL STDUP(NP,NV,NSTD,STSIZE,GSIZE,BFREQ,COEFS,CLASS,PHI,X,
     .             STDVAR)
        IF(TYPE(1:1).EQ.'S') THEN
          CALL SIMLATE(NP,N,ONE,ACTUAL,NSTD,STSIZE,CLASS,PHI,X)
        ELSE
          DO 800 G=1,NMIX
  800       MFREQ(G)=1
          CALL CLASFY(NP,NV,NMIX,MFREQ,CLASS,COEFS,MIXVAR,X)
          CALL REALDAT(NP,ONE,NMIX,CLASS,PHI,X)
        END IF
    1 CONTINUE
      RETURN
  700 FORMAT(//1X,'THE SIZES OF THE STANDARDS ARE ',8I6)
  709 FORMAT(/1X,'MIXED SAMPLE HAS ',I5,' FISH')
      END
C*************************************************************************
      SUBROUTINE SIMLATE(NP,N,NSAMPS,ACT,NSTD,STSIZE,CLASS,PHI,X)
C ARRAYS  X  AND  CLASS CONTAIN LIKELIHOOD AND CLASSIFICATION VALUES
C FOR EVERY FISH IN THE STANDARDS
      INTEGER NP,N,NSAMPS,STSIZE(NP),NSTD,FREQ(1400),CLASS(NSTD),EIGHT,
     .        EST
      REAL ACT(NP),THETA(8,5),PHI(8,8),X(1400,8)
      DOUBLE PRECISION PHIINV(8,8),DET
      EIGHT=8
      DO 801 J=1,NP
        DO 801 JJ=1,NP
  801     PHIINV(J,JJ)=DBLE(PHI(J,JJ))
      CALL MXINT(PHIINV,NP,EIGHT,DET)
      IF(DET.EQ.0.0) THEN
        PRINT *,'THE CLASSIFICATION MATRIX IS SINGULAR!'
        PRINT *,'-NO UNIQUE COMPOSITION ESTIMATE IS POSSIBLE - ABORT'
        STOP
      END IF
      DO 10 LOOP=1,NSAMPS
        CALL SIMMIX(NP,N,NSTD,STSIZE,FREQ,ACT)
        CALL COOKLRD(NP,N,NSTD,FREQ,CLASS,THETA,PHIINV)
        CALL MILLARC(NP,N,THETA,PHI,TYPE)
        CALL ML(NP,N,NSTD,FREQ,THETA,X,TYPE)
        CALL ESTOUT(NP,THETA)
   10 CONTINUE
      RETURN
      END
C*************************************************************************
      SUBROUTINE REALDAT(NP,NSAMPS,NMIX,CLASS,PHI,X)
```

```
C ARRAYS  X  AND  CLASS CONTAIN LIKELIHOOD AND CLASSIFICATION VALUES
C FOR EVERY FISH IN THE MIXED SAMPLE
      INTEGER NP,N,NSAMPS,NMIX,G,CLASS(NMIX),FREQ(1400),DUMMY1(1),
     .        DUMMY2(1),EIGHT,ONE
      REAL PHI(8,8),X(1400,8),THETA(8,5)
      CHARACTER TYPE*10,OPT(2)
      DOUBLE PRECISION PHIINV(8,8),DET
      COMMON /OPTIONS/ TYPE,OPT
      EIGHT=8
      ONE=1
      DO 801 J=1,NP
        DO 801 JJ=1,NP
  801     PHIINV(J,JJ)=DBLE(PHI(J,JJ))
      CALL MXINT(PHIINV,NP,EIGHT,DET)
      IF(DET.EQ.0.0) THEN
        PRINT *,'THE CLASSIFICATION MATRIX IS SINGULAR!'
        PRINT *,'-NO UNIQUE COMPOSITION ESTIMATE IS POSSIBLE - ABORT'
        STOP
      END IF
      IF(OPT(2).EQ.'N') THEN
        DO 800 G=1,NMIX
  800     FREQ(G)=1
        N=NMIX
        CALL COOKLRD(NP,N,NMIX,FREQ,CLASS,THETA,PHIINV)
        CALL MILLARC(NP,N,THETA,PHI,TYPE)
        CALL ML(NP,N,NMIX,FREQ,THETA,X,TYPE)
        CALL ESTOUT(NP,THETA)
      ELSE
        DUMMY1(1)=NMIX
        DUMMY2(1)=NMIX
        N=NMIX
        DO 20 LOOP=1,NSAMPS
          CALL RESAMP(ONE,NMIX,DUMMY1,DUMMY2,FREQ)
          CALL COOKLRD(NP,N,NMIX,FREQ,CLASS,THETA,PHIINV)
          CALL MILLARC(NP,N,THETA,PHI,TYPE)
          CALL ML(NP,N,NMIX,FREQ,THETA,X,TYPE)
          CALL ESTOUT(NP,THETA)
   20   CONTINUE
      END IF
      RETURN
      END
C*************************************************************************
      SUBROUTINE STDUP(NP,NV,NSTD,STSIZE,GSIZE,FREQ,COEFS,CLASS,PHI,X,V)
      INTEGER NP,NV,NSTD,STSIZE(NP),GSIZE(NP),CLASS(1400),FREQ(1400),G
      REAL V(1400,16),COEFS(8,17),PHI(8,8),X(1400,8)
      CALL LDF(NP,NV,NSTD,STSIZE,GSIZE,FREQ,COEFS,V)
C     WRITE(6,701)
C     WRITE(6,702) ((COEFS(I,J),J=1,NV+1),I=1,NP)
      CALL CLASFY(NP,NV,NSTD,FREQ,CLASS,COEFS,V,X)
C     WRITE(6,705)
C     WRITE(6,706) (CLASS(G),G=1,NSTD)
      CALL GETPHI(NP,NSTD,FREQ,CLASS,STSIZE,GSIZE,PHI)
C     WRITE(6,707)
C     WRITE(6,708) ((PHI(I,J),J=1,NP),I=1,NP)
      RETURN
  701 FORMAT(//1X,'THE LDF COEFS ARE:'/)
  702 FORMAT(1X,10F12.3)
  705 FORMAT(//1X,'THE CLASSIFICATION VECTOR IS:'/)
  706 FORMAT(1X,20I4)
  707 FORMAT(//1X,'THE CLASSIFICATION MATRIX IS:'/)
  708 FORMAT(1X,8F12.4)
```

```
      END
C*************************************************************************
      SUBROUTINE READIN(NP,NV,STSIZE,NSTD,VARBLE,UNIT,POPNAM)
C NP  =NO. OF STOCK GROUPS SPECIFIED IN CTRL FILE (FOR008.DAT)
C NPIN=NO. OF STOCK GROUPS FOUND ON FOR007.DAT
C THIS SUBROUTINE CHECKS THAT NP AND NPIN ARE EQUAL.
C*****NB: THIS PROGRAM DOES NOT DO VARIABLE SELECTION.  THE
C*****VARIABLES BELOW WERE PREDETERMINED.
      INTEGER NPIN,NP,NV,STSIZE(NP),NSTD,COUNT,G,UNIT,SPLIT(0:8)
      REAL SUM,SIGMA,VARBLE(1400,16),MEANS(8),SDEVS(8)
      CHARACTER POPNAM(NP)*8
      NPIN=0
      G=1
  100 DO 1 I=1,1401
        COUNT=I
        READ(UNIT,*,END=2) (VARBLE(G,L),L=1,NV)
        G=G+1
        IF(G.GT.1401) THEN
          PRINT *,'SAMPLE IS TOO LARGE  ( > 1400 ) FOR STORAGE'
          STOP
        END IF
    1 CONTINUE
C******* THE NEXT LINE IS NO LONGER REQUIRED (IT IS A CDC CYBER QUIRK)
C*******    2 IF(EOF(UNIT).NE.0) NOUGHT=0
C IF COUNT=1 THEN TWO END'S WERE ENCOUNTERED, INDICATING THE END OF FILE
    2 IF(COUNT-1.GT.0) THEN
        NPIN=NPIN+1
        IF(NPIN.GT.NP) THEN
          WRITE(6,701) NP,NPIN
          STOP
        END IF
        IF(COUNT-1.LT.2) THEN
          PRINT *,'ONLY ONE FISH IN ',POPNAM(NPIN),' STANDARD???'
          STOP
        END IF
        STSIZE(NPIN)=COUNT-1
        GO TO 100
      ELSE
        NSTD=G-1
        IF(NPIN.NE.NP) THEN
          WRITE(6,701) NP,NPIN
          STOP
        END IF
      END IF
C OUTPUT SUMMARY STATISTICS ON THE VARIABLES IN THE STANDARDS
      IF(UNIT.EQ.7) THEN
        WRITE(6,702)
        WRITE(6,703) (POPNAM(J),J=1,NP)
        SPLIT(0)=0
        DO 800 J=1,NP
  800     SPLIT(J)=SPLIT(J-1)+STSIZE(J)
        DO 801 L=1,NV
          DO 802 J=1,NP
            SUM=0.0
            SIGMA=0.0
            DO 803 G=SPLIT(J-1)+1,SPLIT(J)
              SUM=SUM+VARBLE(G,L)
              SIGMA=SIGMA+VARBLE(G,L)*VARBLE(G,L)
  803       CONTINUE
            MEANS(J)=SUM/STSIZE(J)
            SDEVS(J)=SIGMA-STSIZE(J)*MEANS(J)*MEANS(J)
            SDEVS(J)=SQRT(SDEVS(J)/(STSIZE(J)-1))
  802     CONTINUE
        WRITE(6,704) L,(MEANS(J),J=1,NP)
        WRITE(6,705) (SDEVS(J),J=1,NP)
```

```
  801 CONTINUE
      WRITE(6,702)
      END IF
      RETURN
  701 FORMAT(/1X,I2,' STOCK GROUPS SPECIFIED BUT ',I2,' FOUND???')
  702 FORMAT(/1X,79('=')//)
  703 FORMAT(25X,'MEAN AND STD DEV SUMMARY OF VARIABLES'//,
     .       1X,'VAR',8(7X,A8)/)
  704 FORMAT(/1X,I2,9X,8(G10.5,5X))
  705 FORMAT(12X,8(G9.3,TL9,'(',4X,')',9X))
      END
C*************************************************************************
      SUBROUTINE RESAMP(NP,NSTD,GSIZE,STSIZE,FREQ)
      INTEGER G,NP,NSTD,GSIZE(NP),STSIZE(NP),FREQ(NSTD),SPLIT(0:8),
     .        STARTJ
      REAL ORDERS(2001)
      COMMON /UNIFORM/ ORDERS
      SPLIT(0)=0
      DO 800 J=1,NP
  800   SPLIT(J)=SPLIT(J-1)+STSIZE(J)
      DO 801 G=1,NSTD
  801   FREQ(G)=0
      DO 1 J=1,NP
        CALL ORDVEC(GSIZE(J),FLOAT(STSIZE(J)),ORDERS)
        STARTJ=SPLIT(J-1)+1
        DO 2 G=1,GSIZE(J)
    2     FREQ(STARTJ+INT(ORDERS(G)))=FREQ(STARTJ+INT(ORDERS(G)))+1
    1 CONTINUE
      RETURN
      END
C*************************************************************************
      SUBROUTINE LDF(NP,NV,NSTD,STSIZE,GSIZE,FREQ,COEFS,VARBLE)
      INTEGER NP,NV,N,GSIZE(NP),FREQ(NSTD),G,SPLIT(0:8),STSIZE(NP),
     .        SIXTEN,GCOUNT
      REAL COEFS(8,17),MEANS(8,16),VARBLE(1400,16),WORK(8,16)
      DOUBLE PRECISION DET,SIGMA1,SIGMA2,SIGMA(16,16)
      SIXTEN=16
      N=0
      SPLIT(0)=0
      DO 800 J=1,NP
        SPLIT(J)=SPLIT(J-1)+STSIZE(J)
        GCOUNT=0
        DO 801 G=SPLIT(J-1)+1,SPLIT(J)
  801     GCOUNT=GCOUNT+FREQ(G)
C AS A CHECK ON THE PREVIOUS ROUTINES, TEST THAT THESE ARE THE SAME
        IF(GSIZE(J).NE.GCOUNT) THEN
          PRINT *,'CONFLICT OF STANDARD SIZE IN SUBROUTINE  LDF'
          STOP
        END IF
        N=N+GSIZE(J)
  800 CONTINUE
      DO 1 J=1,NP
        DO 1 L=1,NV
          SUM=0.0
          DO 2 G=SPLIT(J-1)+1,SPLIT(J)
    2       SUM=SUM+VARBLE(G,L)*FREQ(G)
          MEANS(J,L)=SUM/GSIZE(J)
    1 CONTINUE
C     WRITE(6,702)
C     WRITE(6,703) ((MEANS(J,L),L=1,NV),J=1,NP)
      DO 3 L=1,NV
        DO 3 LL=1,L
          SIGMA1=0.0
          SIGMA2=0.0
```

```
            DO 4 J=1,NP
      4       SIGMA2=SIGMA2+GSIZE(J)*MEANS(J,L)*MEANS(J,LL)
            DO 5 G=1,NSTD
      5       SIGMA1=SIGMA1+VARBLE(G,L)*VARBLE(G,LL)*FREQ(G)
            SIGMA(L,LL)=(SIGMA1-SIGMA2)/(N-NP)
            SIGMA(LL,L)=SIGMA(L,LL)
      3   CONTINUE
C       WRITE(6,700)
        CALL MXINT(SIGMA,NV,SIXTEN,DET)
        IF(DET.EQ.0.0) THEN
          PRINT *,'THE POOLED COVARIANCE MATRIX IS SINGULAR!'
          PRINT *,'POSSIBLY VARIABLES TOO HIGHLY CORRELATED - ABORT'
          STOP
        END IF
C       WRITE(6,701)
        CALL MXMULT(NP,NV,MEANS,SIGMA,WORK)
        DO 6 J=1,NP
          SUM=0.0
          DO 7 L=1,NV
            SUM=SUM+WORK(J,L)*MEANS(J,L)
            COEFS(J,L)=WORK(J,L)
      7   CONTINUE
          COEFS(J,NV+1)=-SUM/2.0
      6 CONTINUE
        RETURN
C 700 FORMAT(///1X,'POOLED COVARIANCE MATRIX'///)
C 701 FORMAT(///1X,'ITS INVERSE'///)
C 702 FORMAT(///1X,'THE RESAMPLED STANDARD MEANS ARE:'///)
C 703 FORMAT(1X,9F12.4)
        END
C*************************************************************************
        SUBROUTINE SIMMIX(NP,N,NSTD,STSIZE,FREQ,ACTUAL)
C THIS SUBROUTINE SIMULATES A MIXED SAMPLE OF SIZE N FROM A MIXED
C POPULATION HAVING PROPORTIONS  ACTUAL .  SINCE THE MIXED SAMPLE
C FISH ARE ALL FROM THE STANDARD IT IS NECESSARY ONLY TO RETURN
C AN INTEGER VECTOR (FREQ) OF FREQUENCIES.
        INTEGER NP,N,NSTD,STSIZE(NP),FREQ(NSTD),SPLIT(0:8),INDEX,G
        REAL ACTUAL(NP),ASPLIT(0:8),ORDERS(2001),WORK,UPPER,AJ
        COMMON /UNIFORM/ ORDERS
        UPPER=1.0
        ASPLIT(0)=0.0
        SPLIT(0)=0
        DO 800 J=1,NP
          SPLIT(J)=SPLIT(J-1)+STSIZE(J)
          ASPLIT(J)=ASPLIT(J-1)+ACTUAL(J)
    800 CONTINUE
        ASPLIT(NP)=1+1.0E-12
        CALL ORDVEC(N,UPPER,ORDERS)
        DO 801 G=1,NSTD
    801   FREQ(G)=0
        J=1
        AJ=ACTUAL(J)
        DO 1 G=1,N
      2   IF(ORDERS(G).LT.ASPLIT(J)) THEN
C WORK IS ALWAYS BETWEEN ZERO AND ONE
C INDEX RANGES FROM SPLIT(J-1)+1 TO SPLIT(J)
            WORK=((ORDERS(G)-ASPLIT(J-1))/AJ)
            INDEX=INT(WORK*STSIZE(J))+1+SPLIT(J-1)
            FREQ(INDEX)=FREQ(INDEX)+1
            GOTO 1
          END IF
          J=J+1
          IF(J.GT.NP) THEN
            PRINT *,'ERROR IN SUBROUTINE SIMMIX.  PROGRAM STOPPED.'
```

```
          STOP
        END IF
        AJ=ACTUAL(J)
        GOTO 2
    1 CONTINUE
      RETURN
      END
C*************************************************************************
      SUBROUTINE CLASFY(NP,NV,NSTD,FREQ,CLASS,COEFS,VARBLE,X)
      INTEGER G,NP,NV,NSTD,FREQ(NSTD),CLASS(NSTD),MAX
      REAL COEFS(8,17),VARBLE(1400,16),LDFVAL(8),BIGEST,X(1400,8)
      CHARACTER TYPE*10
      COMMON /OPTIONS/ TYPE
      DO 1 G=1,NSTD
        IF(FREQ(G).GT.0.OR.TYPE(1:1).EQ.'S') THEN
          DO 2 J=1,NP
            LDFVAL(J)=0.0
            DO 3 L=1,NV
    3         LDFVAL(J)=LDFVAL(J)+COEFS(J,L)*VARBLE(G,L)
            LDFVAL(J)=LDFVAL(J)+COEFS(J,NV+1)
C IT IS ONLY THE RELATIVE SIZE OF THE LINEAR DISCRIMINANT FUNCTIONS
C THAT IS IMPORTANT SO WE CAN SET  LDFVAL(1) TO ZERO AND TRANSLATE
C THE OTHERS BY THE SAME AMOUNT.
C THE  X  MATRIX STORES THE 'LIKELIHOOD' VALUES THAT ARE REQUIRED BY
C THE DIRECT MAXIMUM LIKELIHOOD METHOD.
    2     CONTINUE
          DO 5 J=2,NP
            LDFVAL(J)=LDFVAL(J)-LDFVAL(1)
            X(G,J)=EXP(LDFVAL(J))
    5     CONTINUE
          LDFVAL(1)=0
          X(G,1)=1.0
          BIGEST=LDFVAL(1)
          MAX=1
          DO 4 J=2,NP
            IF(LDFVAL(J).GT.BIGEST) THEN
              MAX=J
              BIGEST=LDFVAL(J)
            END IF
    4     CONTINUE
        ELSE
          MAX=0
          DO 6 J=1,NP
    6       X(G,J)=0.0
        END IF
        CLASS(G)=MAX
    1 CONTINUE
      RETURN
      END
C*************************************************************************
      SUBROUTINE GETPHI(NP,NSTD,FREQ,CLASS,STSIZE,GSIZE,PHI)
      INTEGER G,NP,NSTD,FREQ(NSTD),CLASS(NSTD),STSIZE(NP),
     .        GSIZE(NP),SPLIT(0:8),CCOUNT(8)
      REAL CHECK,PHI(8,8)
      SPLIT(0)=0
      DO 800 J=1,NP
  800   SPLIT(J)=SPLIT(J-1)+STSIZE(J)
      DO 1 J=1,NP
        DO 801 JJ=1,NP
  801     CCOUNT(JJ)=0
        DO 2 G=SPLIT(J-1)+1,SPLIT(J)
    2     IF(FREQ(G).GT.0) CCOUNT(CLASS(G))=CCOUNT(CLASS(G))+FREQ(G)
        CHECK=0.0
        DO 3 JJ=1,NP
          PHI(JJ,J)=FLOAT(CCOUNT(JJ))/FLOAT(GSIZE(J))
```

```
          CHECK=CHECK+PHI(JJ,J)
    3   CONTINUE
        IF(CHECK.LE.0.9999.OR.CHECK.GE.1.0001) THEN
          PRINT *,'ERROR IN SUBROUTINE GETPHI.  PROGRAM STOPPED.'
          WRITE(6,700) CHECK
          STOP
        END IF
    1 CONTINUE
      RETURN
  700 FORMAT(///'OH NO - VALUE CHECK IS ',F8.4,' <>1.0'///)
      END
C*************************************************************************
      SUBROUTINE COOKLRD(NP,N,NSTD,FREQ,CLASS,THETA,PHIINV)
      INTEGER NP,N,NSTD,FREQ(NSTD),CLASS(NSTD),Y(8),G
      REAL THETA(8,5),A(8),SUM,ATOT,ATHTOT
      DOUBLE PRECISION PHIINV(8,8)
      DO 800 J=1,NP
  800   Y(J)=0
      DO 801 G=1,NSTD
  801   Y(CLASS(G))=Y(CLASS(G))+FREQ(G)
      DO 802 J=1,NP
  802   THETA(J,1)=FLOAT(Y(J))/FLOAT(N)
C
CC NOW CALCULATE CORRECTED ESTIMATES
C
      DO 1 J=1,NP
        SUM=0.0
        DO 2 JJ=1,NP
    2     SUM=SUM+PHIINV(J,JJ)*THETA(JJ,1)
        THETA(J,2)=SUM
    1 CONTINUE
C
CC NOW CALCULATE CONSTRAINED COOK ESTIMATES.  THIS METHOD CAN
C REQUIRE MORE THAN ONE ITERATION, BUT NEVER MORE THAN NP-1
      DO 888 J=1,NP
  888   THETA(J,3)=THETA(J,2)
      DO 10 ITERS=1,NP
        DO 889 J=1,NP
  889     IF(THETA(J,3).LT.0.0) GO TO 9
        RETURN
    9   ATOT=0.0
        ATHTOT=0.0
        DO 11 J=1,NP
          IF(THETA(J,3).GT.0.0) THEN
            A(J)=1.0
          ELSE
            A(J)=0.0
          END IF
          ATOT=ATOT+A(J)
          THETA(J,3)=A(J)*THETA(J,3)
          ATHTOT=ATHTOT+THETA(J,3)
   11   CONTINUE
        DO 12 J=1,NP
   12     THETA(J,3)=THETA(J,3)-A(J)*(ATHTOT-1.0)/ATOT
   10 CONTINUE
      PRINT *,'FAULT WITH CODE FOR COOK''S CONSTRAINED ESTIMATE'
      WRITE(6,700) ITERS,(A(J),J=1,NP)
      WRITE(6,701) ((THETA(J,K),K=1,3),J=1,NP)
  700 FORMAT(1X,'ITERS=',I2,'  A: ',8F4.1)
  701 FORMAT(1X,'THETA:'/8(/1X,3F12.8))
      STOP
      END
C*************************************************************************
      SUBROUTINE MILLARC(NP,N,THETA,X,TYPE)
```

```
C THIS SUBROUTINE OBTAINS THE MAXIMUM LIKELIHOOD CONSTRAINED CORRECTED
C ESTIMATES AS ADVOCATED BY MILLAR.  THE EM (EXPECTATION MAXIMIZATION)
C ALGORITHM IS USED.  SUBROUTINE  EM  BELOW IS THE SAME VERSION USED
C BY WDF TO DO DIRECT MAXIMUM LIKELIHOOD.  THE INPUT FOR  EM  IS
C CREATED BELOW - MUCH OF IT IS SETTING ALGORITHM SPECIFICATIONS.
      INTEGER NP,N,Y(8),IMAX,MAXG,NROWSX
      REAL THETA(8,5),THETA4(8),LAMBDA(8),X(8,8),TOL
      CHARACTER OPT1*3,OPT5*3,TYPE*10
      IMAX=100
C GIVE 'EVEN' STARTING VALUES FOR THE ESTIMATES
      DO 4 J=1,NP
    4   THETA4(J)=1.0/FLOAT(NP)
      TOL=1.0E-06
C OPT1 IS A PRINT OPTION.  IF YES THEN INTERMEDIATE RESULTS ARE PRINTED
C EVERY ITERATION.  IT WILL BE SET TO 'NO' BUT COULD BE ALLOWED AS A
C CONTROL FILE OPTION IF DESIRED.
C OPT5 CONTROLS USE OF THE EM ACCELERATION SUBROUTINE  ACCEL
      OPT1='NO'
      OPT5='YES'
      MAXG=NP
C CALCULATE CLASSIFICATION FREQUENCIES
      DO 1 J=1,NP
    1   Y(J)=NINT(THETA(J,1)*N)
C NROWSX  IS THE NUMBER OF ROWS IN  X  (THE CLASSIFICATION MATRIX)
      NROWSX=8
C
CC HERE GOES.....
C
      CALL EM(NP,IMAX,LAMBDA,X,THETA4,TOL,OPT1,OPT5,MAXG,Y,N,NROWSX)
      DO 2 J=1,NP
    2   THETA(J,4)=THETA4(J)
      RETURN
      END
C*******************************************************************************
      SUBROUTINE ML(NP,N,MAXG,Y,THETA,X,TYPE)
C THIS SUBROUTINE OBTAINS THE DIRECT MAXIMUM LIKELIHOOD ESTIMATES.
C THE EM (EXPECTATION MAXIMIZATION) ALGORITHM IS USED.
C SUBROUTINE  EM  BELOW IS THE SAME VERSION USED BY WDF.
C THE INPUT FOR  EM  IS CREATED BELOW -
C MUCH OF IT IS SETTING ALGORITHM SPECIFICATIONS.
      INTEGER NP,N,Y(MAXG),IMAX,MAXG,NROWSX
      REAL THETA(8,5),THETA5(8),LAMBDA(1400),X(1400,8),TOL
      CHARACTER OPT1*3,OPT5*3,TYPE*10
      IMAX=100
C GIVE 'EVEN' STARTING VALUES FOR THE ESTIMATES
      DO 4 J=1,NP
    4   THETA5(J)=1.0/FLOAT(NP)
      TOL=1.0E-06
C OPT1 IS A PRINT OPTION.  IF YES THEN INTERMEDIATE RESULTS ARE PRINTED
C EVERY ITERATION.
      OPT1='NO'
      OPT5='YES'
C
CC HERE GOES.....
C
      NROWSX=1400
      CALL EM(NP,IMAX,LAMBDA,X,THETA5,TOL,OPT1,OPT5,MAXG,Y,N,NROWSX)
      DO 2 J=1,NP
    2   THETA(J,5)=THETA5(J)
      RETURN
      END
C*******************************************************************************
      SUBROUTINE ORDVEC(NUM,UPPER,ORDERS)
      INTEGER NUM,SEED
      REAL UPPER,ORDERS(NUM+1)
      COMMON /RANDOM/ SEED
      ORDERS(1)=-ALOG(RAN(SEED))
      DO 1 I=2,NUM+1
```

```
    1   ORDERS(I)=ORDERS(I-1)-ALOG(RAN(SEED))
      C=UPPER/ORDERS(NUM+1)
      DO 2 I=1,NUM
    2   ORDERS(I)=ORDERS(I)*C
      RETURN
      END
C*************************************************************************
      SUBROUTINE RESULTS(NP,NS,ACTUAL,TYPE,POPNAM)
      INTEGER NP,NS
      REAL THSUM(8,5),THSSQ(8,5),THMSE(8,5),ACTUAL(NP),COVAR(8,8)
      CHARACTER TYPE*10,POPNAM(NP)*8
      COMMON THSUM,THSSQ,COVAR
      IF(TYPE(1:1).EQ.'A') THEN
        WRITE(6,704)
        NS=0
      ELSE
        WRITE(6,700) NS
      END IF
C THE FOLLOWING CODE CAN SUFFER FROM NUMERICAL INACCURACIES IN EXTREME
C CASES (E.G., IF THE ESTIMATE HAS NEAR ZERO VARIANCE).
C HENCE THE IF STATEMENT TO TEST NON-NEGATIVITY OF THSSQ(J,K) BELOW.
C IN ANY CASE, THE PRINTED VALUES SHOULD BE GOOD TO AT LEAST 3
C DECIMAL PLACES.
      IF(NS.GT.1) THEN
        DO 1 K=1,5
          DO 1 J=1,NP
            THSUM(J,K)=THSUM(J,K)/NS
            THSSQ(J,K)=(THSSQ(J,K)-NS*THSUM(J,K)*THSUM(J,K))/(NS-1)
            IF(THSSQ(J,K).LT.0) THSSQ(J,K)=0
            THSSQ(J,K)=SQRT(THSSQ(J,K))
    1   CONTINUE
        DO 11 K=1,NP
          DO 12 J=1,NP
            COVAR(J,K)=COVAR(J,K)-NS*THSUM(J,5)*THSUM(K,5)
            COVAR(J,K)=COVAR(J,K)/(NS-1)
   12     CONTINUE
          IF (COVAR(K,K).LT.0) COVAR(K,K)=0
   11   CONTINUE
      END IF
      WRITE(6,701)
      DO 888 J=1,NP
  888 WRITE(6,702) POPNAM(J),(THSUM(J,K),K=1,5)
      IF(NS.GT.1) THEN
        WRITE(6,703) NS
        WRITE(6,701)
        DO 889 J=1,NP
  889   WRITE(6,702) POPNAM(J),(THSSQ(J,K),K=1,5)
        IF(TYPE(1:1).EQ.'S') THEN
          DO 2 K=1,5
            DO 2 J=1,NP
              WORK=ACTUAL(J)-THSUM(J,K)
              THMSE(J,K)=SQRT(WORK*WORK+THSSQ(J,K)*THSSQ(J,K))
    2     CONTINUE
          WRITE(6,705)
          WRITE(6,701)
          DO 890 J=1,NP
  890     WRITE(6,702) POPNAM(J),(THMSE(J,K),K=1,5)
        END IF
        WRITE(6,706) NS
        CALL MXOUT(NP,COVAR)
        WRITE(6,707)
        CALL RHO(NP,COVAR)
```

```
        CALL MXOUT(NP,COVAR)
      END IF
      RETURN
  700 FORMAT(//'1'/1X,'TABLE OF COMPOSITION ESTIMATE MEANS.  ',
     .'NUMBER OF RUNS =',I5//)
  701 FORMAT(13X,'RAW',6X,'COOK & LORD',
     .6X,'COOK',8X,'MILLAR',6X,'MAXIMUM'/,
     .34X,'CONSTRAINED',2X,'CONSTRAINED',2X,'LIKELIHOOD'//)
  702 FORMAT(1X,A8,F8.4,4(5X,F8.4))
  703 FORMAT(////,1X,'TABLE OF COMPOSITION ESTIMATE STANDARD ',
     .'DEVIATIONS OVER THE',I5,' RUNS'//)
  704 FORMAT(//'1',1X,'TABLE OF COMPOSITION ESTIMATES'//)
  705 FORMAT(////,1X,'TABLE OF SQRT OF MEAN SQUARED ERRORS'//)
  706 FORMAT('1THE COVARIANCE MATRIX OF THE',I5,' MAXIMUM LIKELIHOOD ',
     . 'COMPOSITION ESTIMATES IS:'///)
  707 FORMAT(////' THE CORRESPONDING CORRELATION MATRIX IS:'///)
      END
C*************************************************************************
      SUBROUTINE MXINT(A,NRR,NRA,DET)
      DIMENSION ISW(96), A(*)
      DOUBLE PRECISION A,DET
C THIS MATRIX INVERSION SUBROUTINE TAKES A  NRA*NRA  SQUARE MATRIX AND
C INVERTS THE UPPER  NRR*NRR  SUBMATRIX

      NA = NRA+1
      ND = NRR*NRA
      DO 1 I=1,NRR
    1 ISW(I) = I
      NDIAG = 1
      NUP = 1
      DET = 1.
      DO 9 I=1,NRR
         KA = I
         AM = 0.
         LA = NDIAG
         DO 2 J=I,NRR
            IF (DABS(A(LA)).LE.AM) GO TO 2
            KA = J
            AM = DABS(A(LA))
    2    LA = LA+1
         IF (AM.GT.1.0E-20) GO TO 3
         WRITE (6,14)
         DET = 0.
         RETURN
    3    IF (KA.EQ.I) GO TO 5
         MUP = I
         DO 4 MLOW=KA,ND,NRA
            S = A(MUP)
            A(MUP) = A(MLOW)
            A(MLOW) = S
    4    MUP = MUP+NRA
         IS = ISW(KA)
         ISW(KA) = ISW(I)
         ISW(I) = IS
    5    AA = A(NDIAG)
         DET = DET*AA
         DO 6 J=I,ND,NRA
    6    A(J) = A(J)/AA
         A(NDIAG) = 1./AA
         NEL = NUP
         DO 8 J=1,NRR
            IF (J.EQ.I) GO TO 8
            AA = A(NEL)
            A(NEL) = 0.
```

```
            KA = I
            DO 7 K=J,ND,NRA
               A(K) = A(K)-AA*A(KA)
 7          KA = KA+NRA
 8       NEL = NEL+1
         NUP = NUP+NRA
 9    NDIAG = NDIAG+NA
      I = 1
10    IF (ISW(I).NE.I) GO TO 11
      I = I+1
      IF (I.GT.NRR) GO TO 13
11    J = ISW(I)
      ISW(I) = ISW(J)
      ISW(J) = J
      MLOW = (I-1)*NRA+1
      MUP = (J-1)*NRA+1
      DO 12 K=1,NRR
         S = A(MLOW)
         A(MLOW) = A(MUP)
         A(MUP) = S
         MLOW = MLOW+1
12    MUP = MUP+1
      GO TO 10

13    RETURN
14    FORMAT (1X,19HPIVOT ZERO IN MXINT)
      END
C*************************************************************************
      SUBROUTINE MXOUT(NP,V)
      DIMENSION V(8,8)
      REAL V
C THIS MATRIX PRINTING ROUTINE IS GENERAL ENOUGH TO PRINT ANY SIZE SQUARE
C MATRIX BY SPLITTING IT UP INTO PAGE SIZE PORTIONS.  THE MATRIX IS
C ASSUMED SYMMETRIC AND ONLY THE LOWER TRIANGULAR PART IS PRINTED.

      DO 39 I=1,NP,7
         K=I+6
         IF(K.GT.NP) K=NP
            WRITE(6,33)I,NP,I,K
            WRITE(6,35)(L,L=I,K)
            DO 37 J=I,NP
            L=J
            IF(J.LE.K)GO TO 1
            L=K
 1          WRITE(6,36)J,(V(J,M),M=I,L)
37       CONTINUE
39       CONTINUE
33    FORMAT(1H-,13H(POPULATIONS ,I2,1H-,I2,3H * ,I2,1H-,I2,1H))
35    FORMAT(2H0 ,7I12)
36    FORMAT(1H ,I3,2X,7E12.4)
      RETURN
      END
C*************************************************************************
      SUBROUTINE RHO(NP,COVAR)
      REAL COVAR(8,8)

      DO 3 I=1,NP
      DO 3 J=NP,I,-1
        IF(COVAR(I,I)*COVAR(J,J).EQ.0.0) THEN
          COVAR(J,I)=0.0
          IF(I.EQ.J) COVAR(I,I)=1.0
        ELSE
          COVAR(J,I)=COVAR(J,I)/SQRT(COVAR(I,I)*COVAR(J,J))
        END IF
 3  CONTINUE
```

```
      RETURN
      END
C*************************************************************************
      SUBROUTINE ESTOUT(NP,THETA)
      INTEGER NP,EST
      REAL THETA(8,5),THSUM(8,5),THSSQ(8,5),COVAR(8,8)
      COMMON THSUM,THSSQ,COVAR
      DO 1 EST=1,5
        DO 1 J=1,NP
          THSUM(J,EST)=THSUM(J,EST)+THETA(J,EST)
          THSSQ(J,EST)=THSSQ(J,EST)+THETA(J,EST)*THETA(J,EST)
    1 CONTINUE
      DO 2 J=1,NP
        DO 2 K=1,NP
    2     COVAR(J,K)=COVAR(J,K)+THETA(J,5)*THETA(K,5)
C     WRITE(2,700) ((THETA(J,EST),J=1,NP),EST=2,5)
C 700 FORMAT(16(1X,F6.4))
      RETURN
      END
C*************************************************************************
      SUBROUTINE MXMULT(N1,N2,A,B,C)
C CALCULATES  C = AB.  A  IS (N1*N2) AND  B IS (N2*N2)
      DIMENSION A(8,16),B(16,16),C(8,16)
      DOUBLE PRECISION B
      DO 1 I=1,N1
         DO 1 J=1,N2
            S=0.0
            DO 2 K=1,N2
    2          S=S+A(I,K)*B(K,J)
            C(I,J)=S
    1    CONTINUE
      RETURN
      END
C*************************************************************************
      SUBROUTINE EM(NP,IMAX,LAMBDA,X,THETA,TOL,OPT1,OPT5,MAXG,Y,N,NROWS)
      REAL LAMBDA(MAXG),X(NROWS,8),THETA(NP),WORK(8),THETA1(8)
      REAL DIFF1(8),DIFF2(8)
      INTEGER G,Y(MAXG),POINTR(1400),NROWS,GCOUNT,MAXG,TOTG
      CHARACTER OPT1*3,OPT5*3,POPNAM(8)*8,FLAG
      DATA POPNAM/'POPN 1','POPN 2','POPN 3','POPN 4','POPN 5','POPN 6',
     .            'POPN 7','POPN 8'/
      LAG=1
      FLAG='N'
      DO 72 K=1,NP
   72   DIFF1(K)=0.0
      HOOD=0.0
      AA=1.0
C CONDENSE  X & Y
      GCOUNT=0
      DO 8 G=1,MAXG
        IF(Y(G).GT.0) THEN
          GCOUNT=GCOUNT+1
          POINTR(GCOUNT)=G
        END IF
    8 CONTINUE
      TOTG=GCOUNT

      DO 1 ITER=0,IMAX
C THE VARIABLES  FLAG AND LAG  CONTROL THE ACCELERATION PROCESS.
C IF FLAG.EQ.'Y' IS TRUE THEN IT INDICATES A SUCCESSFUL ACCELERATION
C IN WHICH CASE  LAMBDA & HOOD  HAVE ALREADY BEEN CALCULATED IN
C SUBROUTINE ACCEL.
        IF(FLAG.EQ.'Y')GO TO 333
        OHOOD=HOOD
        HOOD=0.0
```

```
      DO 2 GCOUNT=1,TOTG
         G=POINTR(GCOUNT)
         LAMBDA(G)=0.0
         DO 10 K=1,NP
            LAMBDA(G)=LAMBDA(G)+X(G,K)*THETA(K)
10       CONTINUE
         IF(LAMBDA(G).LT.1.0E-12.OR.LAMBDA(G).GT.1.0E+12) THEN
           WRITE(6,98)ITER,G,LAMBDA(G)
         END IF
2     CONTINUE
C TEST TO SEE WHETHER IT IS NECESSARY TO CALCULATE THE LOG LIKELIHOOD
C ON THIS ITERATION
      IF(AA.LT.TOL.OR.ITER.EQ.IMAX)GO TO 21
      IF(OPT5.EQ.'NO'.OR.LAG.GE.3.OR.ITER.LE.8)GO TO 334
21    DO 22 GCOUNT =1,TOTG
         G=POINTR(GCOUNT)
         HOOD=HOOD+Y(G)*ALOG(LAMBDA(G))
22    CONTINUE
333   IF(AA.LT.TOL.OR.ITER.EQ.IMAX)GO TO 900
C TEST TO SEE IF PRINTOUT REQUIRED EACH ITERATION
334   IF(OPT1.EQ.'NO') GO TO 14
      WRITE(6,37)ITER
      IF(HOOD.NE.0.0) WRITE(6,38)HOOD
      WRITE(6,48)
      DO 29 K=1,NP
         WRITE(6,39)K,POPNAM(K),THETA(K)
29    CONTINUE
      WRITE(6,40)AA

14    DO 11 K=1,NP
11       WORK(K)=0.0
      DO 3 GCOUNT=1,TOTG
         G=POINTR(GCOUNT)
         QUOTI=Y(G)/LAMBDA(G)
         DO 3 K=1,NP
            WORK(K)=WORK(K)+QUOTI*X(G,K)
3     CONTINUE
      DO 4 K=1,NP
         THETA1(K)=WORK(K)*THETA(K)/N
         DIFF2(K)=DIFF1(K)
         DIFF1(K)=THETA1(K)-THETA(K)
4     CONTINUE
      AA=0.0
      DO 20 K=1,NP
         AA=AMAX1(AA,ABS(DIFF1(K)))
         THETA(K)=THETA1(K)
20    CONTINUE
      IF(OPT5.EQ.'NO'.OR.ITER.LT.10)GO TO 1
      LAG=LAG-1
      FLAG='N'
      IF(LAG.NE.0)GO TO 1
      HOOD=HOOD+2*(HOOD-OHOOD)
      CALL ACCEL(NP,MAXG,DIFF2,DIFF1,THETA,HOOD,LAG,AA,X,LAMBDA,Y,
                 NROWS,POINTR,TOTG)
      IF(LAG.EQ.3) THEN
         IF(OPT1.EQ.'YES') WRITE(6,50) ITER
         FLAG='Y'
      END IF
      IF(LAG.EQ.10.AND.OPT1.EQ.'YES')WRITE(6,51)ITER
1  CONTINUE
C.....END OF EM ALGORITHM.....
900   IF(OPT1.NE.'YES') RETURN
      WRITE(6,37)ITER
```

```
      WRITE(6,38)HOOD
      WRITE(6,48)
      DO 30 K=1,NP
         WRITE(6,39) K,POPNAM(K),THETA(K)
   30 CONTINUE
      WRITE(6,40)AA
      IF(AA.LT.TOL) THEN
         WRITE(6,42)
      ELSE
         WRITE(6,41)
      END IF
      RETURN
   37 FORMAT(///1H ,8('=========='),//,1H ,'ESTIMATED COMPOSITION OF ',
     .'MIXED FISHERY AT ITERATION ',I4)
   38 FORMAT(//1H ,'THE LOGLIKELIHOOD IS ',F12.4)
   48 FORMAT(1H //,10X,10HPOPULATION,10X,12HCONTRIBUTION/)
   39 FORMAT(1H ,I6,4X,A8,12X,F9.6)
   40 FORMAT(//1H ,'CONVERGENCE ',F10.6)
   41 FORMAT(3(/1H ,'*****MAXIMUM NUMBER OF ITERATIONS REACHED*****'))
   42 FORMAT(3(/1H ,'*****CONVERGENCE TO SPECIFIED TOLERANCE******'))
   50 FORMAT(//1H ,'************************************',
     .        /1H ,'*SUCCESSFUL ACCELERATION COMPLETED*  ITER=',I4,
     .        /1H ,'************************************')
   51 FORMAT(//1H ,'************************************',
     .        /1H ,'*ACCELERATION ATTEMPTED BUT DECLINED*  ITER=',I4,
     .        /1H ,'************************************')
   98 FORMAT(1H ,'CAUTION: AT ITER ',I4,' LAMBDA(',I4,')=',E9.2)
      END
C*************************************************************************
      SUBROUTINE ACCEL(NP,MAXG,DIFF2,DIFF1,THETA,HOOD,LAG,AA,X,LAMBDA,Y,
     .                 NROWSX,POINTR,TOTG)
      REAL LAMBDA(MAXG)
      INTEGER G,GCOUNT,NROWSX,TOTG,Y(MAXG),POINTR(TOTG),MAXG
      DIMENSION DIFF2(8),DIFF1(8),THETA(8),X(NROWSX,8),RATIO(8),
     .HALT(8),THETA1(8)
      ALPHA=100.0
      SUM=0.0
      DO 1 K=1,NP
         IF(ABS(DIFF1(K)).GE.AA/20) GO TO 2
         HALT(K)=0.0
         GO TO 1
    2    IF(DIFF1(K).LT.0.0)HALT(K)=-THETA(K)/DIFF1(K)
         IF(DIFF1(K).GT.0.0)HALT(K)=(1-THETA(K))/DIFF1(K)
         HALT(K)=INT(HALT(K))
    1 CONTINUE
      DO 3 K=1,NP
         IF(ABS(DIFF1(K)).LT.AA/20)GO TO 3
         RATIO(K)=DIFF1(K)/DIFF2(K)
         ALPHA=AMIN1(ALPHA,RATIO(K))
    3 CONTINUE
      IF(ALPHA.GT.0.75)GO TO 8
      LAG=10
      RETURN
    8 IF(ALPHA.EQ.1.0)ALPHA=0.99
      ALPHA=AMIN1(10.0,(1/(1-ALPHA))-1)
      DO 5 K=1,NP
         THETA1(K)=THETA(K)+AMIN1(HALT(K),ALPHA)*DIFF1(K)
         SUM=SUM+THETA1(K)
    5 CONTINUE
      DO 7 K=1,NP
    7    THETA1(K)=THETA1(K)/SUM
```

```
C CHECK TO SEE WHETHER ACCELERATION GIVES IMPROVEMENT IN LIKELIHOOD
      HOOD1=0.0
      DO 30 GCOUNT=1,TOTG
         G=POINTR(GCOUNT)
         LAMBDA(G)=0.0
         DO 31 K=1,NP
   31       LAMBDA(G)=LAMBDA(G)+X(G,K)*THETA1(K)
         HOOD1=HOOD1+Y(G)*ALOG(LAMBDA(G))
   30 CONTINUE
      IF(HOOD1.GT.HOOD)GO TO 32
      LAG=10
      RETURN
   32 AA=0.0
      HOOD=HOOD1
      DO 33 K=1,NP
         AA=AMAX1(AA,ABS(THETA1(K)-THETA(K)))
         THETA(K)=THETA1(K)
   33 CONTINUE
      LAG=3
      RETURN
      END
```